Resonance used for frequency detection

by Alexander Mlazev

I will describe the method used in my Chord Detector App from 2019 and few other products I have made as Chromatic Tuners and audio effects during the years.

The method is interesting as it is tot using FFT to detect frequencies but a self resonating sensors.

Frequency in mathematics is self repeating of a signal and that is almost never found in pure form in real sounds. At the same time humans perceive the sounds as tonally different and can easily match them to artificially created frequency defined signals. That is because our ears detect sounds by tiny self resonating sensors.

In fact it is very hard for us to distinguish wave forms but it is easy to distinguish tones. If we take a repeating signal with some waveform, we can not hear changes in the place or order of the peaks inside the waveform but we will hear changes in the width of these peaks and perceive that as timbre. We can not even hear if the peaks are moving dynamically inside this waveform without changing widths. No just phase but order and places of the peaks inside the waveform. (You can play with waveforms with my Waveform Sound Generator https://play.google.com/store/apps/details?id=com.bialamusic.waveformsoundgenerator)

So I found out that FFT is not perfect model to work with sounds and started working with self resonating sensors to mimic our perception of sound.

In my method I am using an array of self resonating sensors (Kolbichki). Every sensor is set to work at specific sample length defining its resonance frequency and has a resonance coefficient - _decreaseK and a history array with the same exact sample length to keep its wave form.

Input is handled sample by sample and every sample is handled by every resonator sensor by this formula:

```
newV = inputV + (history[_offset] * decreaseK);
history[_offset] = newV;
_offset += 1;
if (_offset >= history.length)
{
   offset = 0;
}
```

For every sensor then the current resonance value is measured by finding the max positive value inside its history array, the average positive value in the history array and the exact offset of max positive value inside its history array. The positions of the values of the natural harmonics for the max positive value inside the history array (the wave form) of the resonator sensor are found according the position of the max positive value. Their values are multiplied by selected decreasing coefficients and

deducted by the max positive value. That is made to make the output of the long wave length resonator sensor closer to the output of short wavelength sensors.

Then the max value is normalized according the current volume level of the input signal.

Those values are then used as input for my Chord detection algorithm in Chord Detector App from 2019 which you can check here:

https://play.google.com/store/apps/details?id=com.bialamusic.resonancechorddetector

I am also using those values as input for varios other algorithms and apps like Chromatic Tuner:

https://play.google.com/store/apps/details?id=com.bialamusic.chromatictuner

And audio effects:

https://apps.apple.com/us/app/sirens-chor-synth-chorus/id1497361812